NOTE: X+ represents the closure of attribute set X

Question 1:

a)
A->ACF->ACFH->ACDEFGH->ABCDEFGH => pass
BCG->BCDG => VIOLATE BCNF
CF->ACFH->ACDEFGH->ACBDEFGH => pass
D->BD => VIOLATE BCNF
H->DEGH->BDEGH => VIOLATE BCNF

therefore, BCG->D, D->B, and H->DEG violate BCNF

b)
decompose R using FD D->B:
   R1 = D+ = BD
   R2 = R - (R1-D)
     = ABCDEFGH - B
     = ACDEFGH

project FD's onto R1 = BD:
   B+ = B => nothing
   D+ = BD => D->B; D a superkey of R1
   BD+ trivial
   R1 satisfies BCNF with FD: D->B FINAL

project FD's onto R2 = ACDEFGH:
   A is a superkey
   C+ = C => nothing
   D+ = B => nothing
   E+ = E => nothing
   F+ = F => nothing
   G+ = G => nothing
   H+ = BDEGH => violates BCNF; H->DEG
   must decompose R2 further

decompose R2 using H->DEG:
   R3 = H+ = DEGH
   R4 = R2 - (R3 - H)
     = ACDEFGH - (DEGH - H)
     = ACFH

project FD's onto R3 = DEGH:
   D+ = BD => nothing
   E+ = E => nothing
   G+ = G => nothing

H+ = DEGH => H->DEG; H a superkey of R3
DE+ trivial
EG+ trivial
DG+ trivial
DEG+ trivial
no need to further check supersets of H
R3 satisfies BCNF with FD: H->DEG FINAL

project FD's onto R4 = ACFH:
A->CFH; A is a superkey R4
C+ = C => nothing
F+ = F => nothing
H+ = DEGH => nothing
no need to further check supersets of A
CF+ = ACFH => CF->AH; CF a superkey of R4
FH+ trivial
CH+ trivial
no need to further check supersets of CF
R4 satisfies BCNF with FD: A->CFH, CF->AH FINAL

Final lossless decomposition of R follows:
R1=BD; FD: D->B
R3=DEGH; FD: H->DEG
R4=ACFH; FD: A->CFH, CF->AH

Question 2:

a)
Look at all LHSs (AB, B, BCD, BCDE, BCE, DF)

B->CEF => BCEF
(BCE)F->(BCE)DF => BCDEF
(BCDE)F -> (BCDE)AF => ABCDEF
Therefore, B is a key for R, and any superset of B cannot be a key

the only LHS without B is DF:
DF->C => CDF
Therefore DF is NOT a key for R

Therefore, the only key of R is: B

b)
To find the minimal basis for S, we first break all FD's into singleton's:
1) AB -> E

2) AB -> F
3) B -> C
4) B -> E
5) B -> F
6) BCD -> A
7) BCD -> F
8) BCDE -> A
9) BCE -> D
10)DF -> C

Let's call the above set S1. Now sequentially eliminate redundancies by checking closure(LHS[i]) without FD[i]

	FD's ignored	closure(LHS)	result
1) AB -> E	1	ABCE..	discard
2) AB -> F	1, 2	ABCEF..	discard
3) B -> C	1, 2, 3	can't get C	keep
4) B -> E	1, 2, 4	can't get E	keep
5) B -> F	1, 2, 5	BCEDF..	discard
6) BCD -> A	1, 2, 5, 6	BCDEA..	discard
7) BCD -> F	1, 2, 5, 6, 7	can't get F	keep
8) BCDE -> A	1, 2, 5, 6, 8	can't get A	keep
9) BCE -> D	1, 2, 5, 6, 9	can't get D	keep
10)DF -> C	1, 2, 5, 6, 10	can't get C	keep

The resultant S2 is now
1) B -> C
2) B -> E
3) BCD -> F
4) BCDE -> A
5) BCE -> D
6) DF -> C

Reduce LHS of S2 which have multiple attributes:
3) BCD -> F: B+ = BCEDF => reduce 3) to B->F
4) BCDE -> A: B+ = BCEDA => reduce 4) to B->A
5) BCE -> D: B+ = BCED => reduce 5) to B->D
6) DF -> C: D+=D;F+=F => remain

Thus, we have the following sorted, minimal basis for S:
1) B -> A
2) B -> C
3) B -> D
4) B -> E
5) B -> F
6) DF -> C

c)
Following the 3NF synthesis algorithm, we generate a relation per FD
AFTER merging the RHS of the minimal basis, which results the following:
1) B -> ACDEF
2) DF -> C

Thus the set of relations follow:
R1(A,B,C,D,E,F)
R2(C,D,F)

Clearly the attribute set of R2 is contained in R1, thus we can skip R2.
Finally resulting in a single relation:
R1(A,B,C,D,E,F)

d)
To check whether the schema above allows redundancy, we check whether all FD's satisfies
the BCNF. If all FD's satisfy BCNF i.e. all LHS are superkeys, then no redundancies are allowed.

By inspection, it is clear that the FD: DF->C does not satisfy BCNF, since DF+ = DFC, not
superkey of R1(A,B,C,D,E,F). Thus the above schema does NOT satisfy BCNF and allows redundancy.